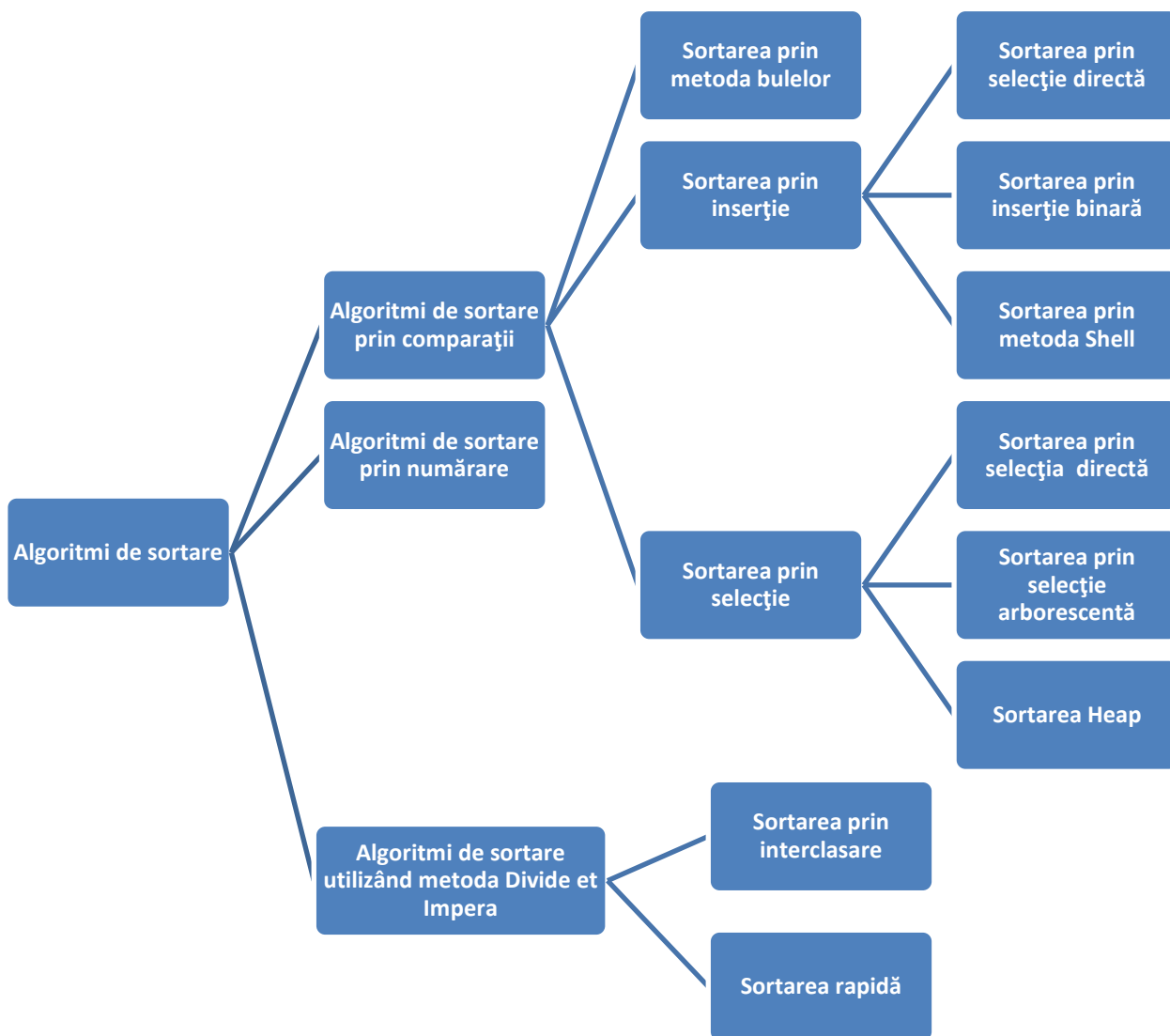


# METODE DE SORTARE

Autor: prof. Neli Seciță

Una dintre cele mai importante probleme de programare atât din punct de vedere teoretic cât și practic este sortarea datelor, și anume rearanjarea obiectelor într-o ordine crescătoare sau descrescătoare.

Sortarea este o operație fundamentală de informatică (multe programe o folosesc ca pas intermediar) și ca urmare, a fost dezvoltat un număr mare de algoritmi de sortare. Care algoritm este cel mai bun pentru o aplicație dată depinde de numărul de obiecte care trebuie sortate, de gradul în care aceste obiecte sunt deja sortate într-un anumit fel și de tipul de mediu electronic care urmează să fie folosit: memoria principală, discuri sau benzi magnetice.



**Sortarea prin metoda bulelor** este una dintre cele mai cunoscute și utilizate metode de ordonare, dar eficiența acesteia este foarte scăzută. Această metodă este eficientă pentru un număr mic de elemente (mai puțin de 15), dar nu pentru vectori cu un număr mare de elemente. Nu necesită multă memorie dar este de două ori mai lentă decât metoda de sortare prin inserție directă în aproape orice situație. Este recomandată în cazurile cu puține elemente de sortat și în cazul în care memoria este limitată. Complexitatea acestui algoritm este de  $O(n^2)$ .

S-a încercat îmbunătățirea acestuia. Una din metodele găsite a fost "sortarea shaker" în care trecerile alternate merg în direcții opuse. Numărul mediu de comparații fiind ușor redus. Și cu această îmbunătățire algoritmul nu se îmbunătățește în așa fel încât să devină mai eficient decât inserția directă.

**Sortarea prin inserție directă** are tot complexitatea  $O(n^2)$ , dar este de peste 2 ori mai eficientă decât algoritmul prin metoda bulelor.

**Sortarea prin metoda Shell** este cel mai rapid algoritm din clasa celor de complexitate secvențială  $O(n^2)$ . Viteza algoritmului este aproximativ de cinci ori mai mare decât sortare prin metoda bulelor și de peste două ori mai mare decât metoda de sortare prin inserție. Algoritmul este mai lent în comparație cu metoda de sortare rapidă, metoda de sortare prin interclasare sau metoda de sortare Heap, dar simplitatea acestuia îl face să fie o foarte bună alegere pentru cazurile în care se dorește sortarea repetată a unor vectori de dimensiune moderată.

**Sortarea prin metoda selecției** și sortarea prin metoda bulelor au aceeași complexitate, și anume  $O(n^2)$ , însă performanțele sortării prin metoda selecției sunt cu 60% mai bune decât sortarea prin metoda bulelor.

**Sortarea Heap** este cel mai slab algoritm din clasa  $O(n \cdot \log_2 n)$ . Nu este cu mult mai slab decât algoritmi din familia Quick Sort și are avantajul față de aceștia că nu este recursiv.

Algoritmii recursivi rulează rapid, dar consumă o cantitate de memorie, ceea ce nu le permite să sorteze tablouri de dimensiuni oricât de mari.

Sortarea Heap este un algoritm care „împacă” viteza cu consumul relativ mic de memorie.

**Sortare prin interclasare** are complexitatea  $O(n \cdot \log_2 n)$ . Deși acest algoritm este stabil și necesită cu 30% mai puține comparații decât sortarea rapidă, metoda este mai lentă decât sortarea rapidă datorită faptului că efectuează mai multe interschimbări. Dezavantajul metodei de sortare prin interclasare este că necesită un vector auxiliar de mărimea originalului. Problema apare când vectorul este mare și spațiu este un factor critic.

**Sortarea rapidă** este unul dintre cei mai rapizi algoritmi de sortare. Acesta reprezintă cea mai bună alegere în toate situațiile în care viteza este un factor foarte important, indiferent de dimensiunea seturilor de date ce se doresc sortate. Cu toate acestea nu trebuie scăpat din vedere faptul că algoritmul nu este stabil iar performanțele acestuia sunt modeste atunci când operează asupra unor liste aproape ordonate.